DIGITAL SIGNATURES

- RSA signatures
- El Gamal Signatures
- DSS

Digital signature

Sign a message w

$Sig(w)$

$(w, Sig(w))$

1.) Everyone is able to verify the message was signed by the correct user → doable with the public key

2.) Only the correct user can sign messages
→ doable with the private key

RSA Signatures

Elements: p, q - large primes, $n = p \cdot q$, e, d

$e = d^{-1} \bmod \phi(n)$

$\varphi(n)$ ←

Euler's teta function

$$\varphi(n) = (p-1)(q-1)$$

PRIVATE: d, p, q

PUBLIC: e, n

SIGNATURE OF w: $\quad Sig(w) = w^d \bmod n$

Verification of $(w, Sig(w))$ $\quad w = (Sig(w))^e \bmod n$

$\boxed{\text{How to fake a signature?}}$

1.) Factorize n
2.) Calculate $\varphi(n)$
3.) Invert e **(RSA problem)**
4.) **from w and $w^d \bmod n$**
(discrete log problem)

All (computationally) hard

How to break a signature scheme

Existential forgery: There exists a message w for which signatures are easy (computationally) to calculate

Universal forgery: All messages can be signed (efficiently) by the adversary

RSA existential forgery

Given pair (w, s) can we create more valid pairs?

(w^2, s^2)

$$Sig(w^2) = (w^2)^d = (w^d)^2 = s^2 \mod n$$

Hash functions

$h: I \longrightarrow K \qquad |I| >> |K| \approx 320$ bits number

Cryptographic hash functions:

1.) it is hard to invert h: given $k \in K$ it is (computationally) hard to find $i \in I$, s.t. $h(i) = k$.

2.) it is hard to find collisions: it is (computationally) hard to find $i_1, i_2 \in I$, s.t. $h(i_1) = h(i_2)$

$[w, h(w), sig(h(w))]$

1. Advantage ⇝ Signatures need to be calculated only for small messages (320-bit)

2. Advantage ⇝ Security!

$[w, h(w), sig(h(w))]$

$[w', h(w)^2, sig(h(w))^2]$

In order to use the existential forgery described above, the adversary needs to find w', s.t. $h(w') = h(w)^2$.
This is computationally hard, because h is a cryptographic hash function.

El Gamal signatures

Elements: p - large prime

q - primitive element of $\mathbb{Z}_p^*$

x - $0 < x < p-1$

$y = q^x \bmod p$

PUBLIC: p, q, y

PRIVATE: x

To sign message w: 1.) choose randomly $r \in \mathbb{Z}_{p-1}^*$

→ Multiplicative group mod $p-1$

$a^{-1} \bmod p-1$ exists iff $\gcd(a, p-1) = 1$

$\mathbb{Z}_{p-1}^*$ - set of all invertible elements mod $p-1$

- set of all a coprime to $p-1$

2.) $a = g^r \mod p$

3.) $b = r^{-1} \cdot (w - a \cdot x) \mod (p-1)$

Verification of $(w, (a,b))$

$$g^w \stackrel{?}{\equiv} y^a a^b \mod p$$

$$\equiv (g^x)^a \cdot (g^r)^b \mod p$$

$$\equiv g^{ax} \cdot g^{\overbrace{r[r^{-1}}^{=1} \cdot (w-ax)]} \mod p$$

$$\equiv g^{ax} \cdot g^w \cdot g^{-ax} \mod p$$

$$\equiv g^w \mod p$$

Vulnerabilities of El gamal signature (ex 2)

1.) There is an existential forgery on El Gamal signatures, that doesn't use any valid message - signature pair

2.) Given $(w, (a,b))$ it is possible to find signatures for $w' = \lambda (w - \beta b) \mod p-1$ for arbitrarily chosen $\beta \in \mathbb{Z}_p^*$ and $\lambda = g^\beta \mod p$

3.) Given (w_1, a_1, b_1) and (w_2, a_1, b_2) it is possible to efficiently calculate x, thus completely breaking the scheme efficienty calculate x, thus completely breaking the scheme

to solve for X

not necessarily a prime

$$ax \equiv b \mod n$$

1.) $\gcd(a,n) = 1$, $\Rightarrow$ Extended Euclid's algorithm to find $a^{-1} \mod n$ + multiply both sides with a^{-1}. (a.k.a. divide by a)

2.) $\gcd(a,n) = k$ $\wedge$ k does not divide b $\Rightarrow$ No solution

3.) $\gcd(a,n) = k$ $\wedge$ $k \mid b$ $=$ k solutions

Algorithm: Solve

$$\frac{a}{k} x \equiv \frac{b}{k} \mod \frac{n}{k}$$

NOTE: $\gcd\left(\frac{n}{k}, \frac{a}{k}\right) = 1$

Solution: $x = s$

Solutions to $ax \equiv b \mod n$

are of the form $s + i\frac{n}{k}$ for $i \{0, \dots, k-1\}$

Example:

$10x \equiv 5 \mod 15$ $\qquad k = \gcd(10, 15) = 5$

1.) Solve $2x \equiv 1 \mod 3$

$x = 2$

2.) Solutions of the original problem

$2 + i \cdot 3$ for $i \in \{0, 1, 2, 3, 4\}$

$x \in \{2, 5, 8, 11, 14\}$ //

DSA - digital signature algorithm

Why not ElGamal?

1.) Size of the signature
$a \mod p$ l-bits
$b \mod p-1$ l-bits

2.) both a and p appear in the exponent of the verification algorithm. This computationally expensive.

Elements: p - large prime l-bit $(512 \leq l \leq 1024)$
l - $64k$

q - 160-bit prime s.t. $q \mid (p-1)$

$r = h^{\frac{(p-1)}{q}} \mod p$, h is a primitive element of $\mathbb{Z}_p^*$

$r = h^{\overline{\ q}} \bmod p$, h is a primitive element of $\mathbb{Z}_p^*$

$\{h, h^2, \ldots, h^{p-1}\} = \{1, \ldots, p-1\}$

$\{r, r^2, \ldots, r^q\}$

$(r^q) = \left(h^{\frac{p-1}{q}}\right)^q = h^{p-1} \equiv 1 \bmod p$ (Fermat's little thm)

r is an element of $\mathbb{Z}_p^*$ of order q

$r^k \equiv r^{k \bmod q} \bmod p$

$$\underbrace{r\cdot r\cdot r\cdot \;\cdots\; r}_{\substack{q\\ \shortparallel\\ 1}},\ \underbrace{r\ r\ r\ r\ r\ \cdots}_{q},\ \underbrace{\qquad\qquad}_{r \bmod q}$$

Private: $0 < x < q$

Public: $p, q, g = r^x \bmod p, r$

to sign w: 1.) Choose a random $k \in \mathbb{Z}_q^*$

$a = (r^k \bmod p) \bmod q$

$b = k^{-1}(w + ax) \bmod q$

inverse mod q

Verification:

1.) $z \equiv b^{-1} \bmod q$

2.) $u_1 \equiv w \cdot z \bmod q$

$u_2 \equiv a \cdot z \bmod q$

3.) $$\left(r^{u_1} \cdot y^{u_2} \bmod p\right) \bmod q \stackrel{?}{\equiv} a$$

$$\left(r^{w \cdot z} \cdot \left(r^{x}\right)^{a \cdot z} \bmod p\right) \bmod q$$

$$\left(r^{w \cdot z + a \cdot x \cdot z} \bmod p\right) \bmod q$$

$$\left(r^{z(w+ax)} \bmod p\right) \bmod q$$

$$\left(r^{k(w+xa)^{-1} \cdot (w+ax)} \bmod p\right) \bmod q \qquad (a^{-1} \cdot b)^{-1} = a b^{-1}$$

$$\left(r^{k} \bmod p\right) \bmod q) \equiv a$$

SUBLIMINAL CHANNELS

Note that El Gamal (DSA, OSS) use two random numbers to calculate the signature.

HW

use two random numbers to calculate the signature

random r

private x

if x is shared with another user, r can be used to send secret messages

$$b = r^{-1}(w - ax) \mod p-1$$

Solve for $(w-ax)$ (either $\gcd((w-ax), p-1) = 1$

or $b \mid \gcd(w-ax, p-1)$)

In the second case r^{-1} has $\gcd(w-ax, p-1)$ solutions

the secret message fullfills $g^r \equiv a \mod p$.

Chaum blind signatures
Lamport 1-time signature
$\Big\}$ In slides